Kapitel 4:
Steuerung von Softwareprojekten

4.1 Methoden der Projektsteuerung

4.2 Ausgewählte Projektmetriken

4.3 Berichterstattung

4.4 Produktivitätssteigerung

Projektsteuerung
Aufgaben, Mindestvoraussetzungen

Zentrale Aufgaben der Projektsteuerung: Den aktuellen Projektstatus sichtbar machen, um Plan-Abweichungen rechtzeitig erkennen und Korrekturmaßnahmen ergreifen zu können.

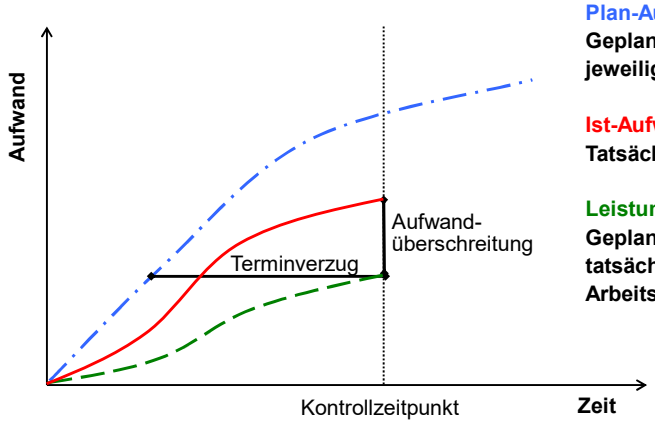
Mindestvoraussetzungen für Projektsteuerung

- akzeptierte Projektpläne, insbesondere hinsichtlich Aufwandsschätzungen und Termine
- Verantwortlichkeiten für alle Projektaufgaben (PL, KM, QS, …) festgelegt
- geeignetes Prozessmodell
- geschultes Personal
- definierte Steuerungsmethode(n)
- festgelegte Zeitpunkte (z.B. wöchentlich, monatlich, zu Meilensteinen, bei Bedarf)
- Visualisierung

Ausgewählte Steuerungsmethoden
Earned Value-Analyse (Aufwand-Termin-Treue) (i)

Eigenschaften:	• **Kontinuierlicher Vergleich des/der aufgelaufenen Aufwands/Kosten mit dem/den für die erbrachte Leistung geplanten Aufwand/Kosten.** • **Ableiten der beiden Steuergrößen: Aufwands-/Kosten-abweichung, Terminabweichung**
Anwendungs-voraussetzungen:	• **Das Projekt in Arbeitspakete inklusive Termine und Aufwände fein strukturieren.** • **Aufwandserfassung je Arbeitspaket** • **Geeignete Toolunterstützung**
Vor-/Nachteile:	☺ **sehr aussagekräftig** ☺ **Lohnt sich vor allem bei größeren Entwicklungen** ☺ **während der gesamten Projektlaufzeit einsetzbar** ☹ **Aufwand für granulare Planung und Aufwandserfassung**

Ausgewählte Steuerungsmethoden
Earned Value-Analyse (Aufwand-Termin-Treue) (ii)



Plan-Aufwand:
Geplanter Aufwand zu dem jeweiligen Zeitpunkt

Ist-Aufwand:
Tatsächlich erbrachter Aufwand

Leistungswert (Earned Value):
Geplanter Aufwand für die tatsächlich erbrachte Arbeitsleistung (Soll-Aufwand)

Bisher weniger Aufwand aufgelaufen als geplant (scheinbar alles ok).

Das Projekt ist aber stark in Terminverzug; der Aufwand wurde überschritten

Ausgewählte Steuerungsmethoden
Steuerung durch Integrationsstufen (i)

Eigenschaften:	• **Verfolgung des Fortschritts von Arbeitspaketen (Fertigstellungsgrad) vom Feinentwurf bis zum Beginn des Systemtests** • **Verfolgung der Termine und Aufwände der Integrationsstufen (weniger der der Arbeitspakete)**
Anwendungs-voraussetzungen:	• **Alle Entwicklungsrelevanten Arbeitspakete liegen vor** • **Zuordnung der Arbeitspakete zu den Leistungsmerkmalen**
Vor-/Nachteile:	☺ **Geringer Aufwand** ☺ **Fortschrittskontrolle anhand von objektiven Kriterien** ☺ **Verfolgung großer Arbeitspakete leicht möglich (über Integrationsstufen).** ☹ **Nicht für alle Projektphasen geeignet**

Ausgewählte Steuerungsmethoden
Steuerung durch Integrationsstufen (ii)

	Integrationsstufe 1			Integrationsstufe 2		…
	LM1	LM2	LM3	LM4	LM5	…
AP1	X		X	X		
AP2	X				X	
…						

AP: Arbeitspaket
LM: Leistungsmerkmal

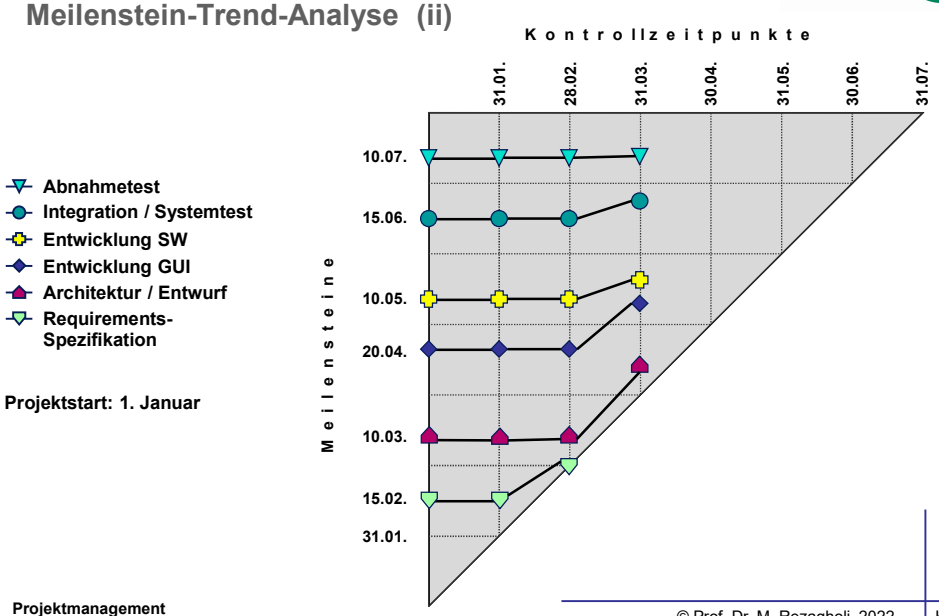
Ausgewählte Steuerungsmethoden
0-100-Methode (Mikro-Meilenstein-Methode)

Prinzip:	• Nur fertige Arbeitspakete werden registriert • Im Vordergrund steht die Verfolgung des Fertigstellungsgrads
Anwendungs-voraussetzungen:	• Kleine Arbeitspakete definieren (Arbeitspakete als sehr kleine Meilensteine) • Projektteam von der Methode überzeugen
Vor-/Nachteile:	☺ Objektiv ☺ Wirkt motivationssteigernd ☹ Hohe Disziplin erforderlich ☺ Auch geeignet für Krisenprojekte

Ausgewählte Steuerungsmethoden
Meilenstein-Trend-Analyse (i)

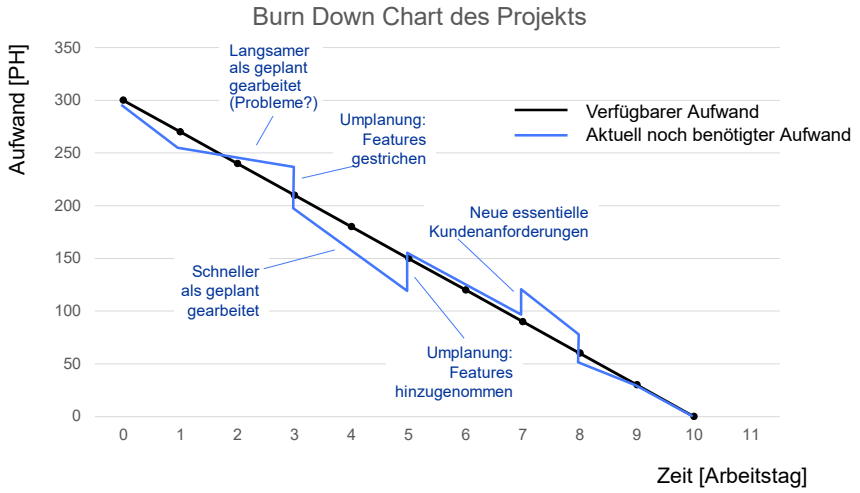
Prinzip:	• Wenige große Meilensteine (Arbeitspakte als Meilensteine) • Regelmäßige Neuabschätzung des Plantermins je Meilenstein • Darstellung der zeitlichen Entwicklung der Meilensteine
Anwendungs-voraussetzungen:	• Präzise definierte Meilensteine • „Mut“ der Schätzer
Vor-/Nachteile:	☺ Wenig Aufwand ☹ Rein subjektive Einschätzung ☹ Plan-Plan-Vergleich

Ausgewählte Steuerungsmethoden
Meilenstein-Trend-Analyse (ii)



Projektstart: 1. Januar

Ausgewählte Steuerungsmethoden
Burn-Down-Chart



Experten-Review

Erfahrungsbasierte Bewertung des Projektstatus durch Experten, um Abweichungen von geplanten Resultaten festzustellen und geeignete Maßnahmen zu definieren.

Kapitel 4:
Steuerung von Softwareprojekten

4.1 Methoden der Projektsteuerung
4.2 Ausgewählte Projektmetriken
4.3 Berichterstattung
4.4 Produktivitätssteigerung

Ausgewählte Projektmetriken 1/6

Leistungsproduktivität

$$\mathrm{LP} = \frac{\text{Funktionaler Umfang}}{\text{Aufwand}}$$

Project Delivery Rate

$$\mathrm{PDR} = \frac{\text{Aufwand}}{\text{Funktionaler Umfang}}$$

Kostenproduktivität

$$\mathrm{KP} = \frac{\text{Funktionaler Umfang}}{\text{Projektkosten [€]}}$$

Project Cost by Delivery

$$\mathrm{SK} = \frac{\text{Projektkosten [€]}}{\text{Funktionaler Umfang}}$$

Funktionaler Umfang, gemessen in Function Points (FP), Source Lines of Code (SLOC), …
Aufwand, gemessen in Personenmonaten (PM), Personenwochen (PW), …

Ausgewählte Projektmetriken 2/6

Projektaufwandsverteilung

Liefergeschwindigkeit

$$\mathrm{LG} = \frac{\text{Funktionaler Umfang}}{\text{Projektdauer}}$$

Fluktuation

$$\mathrm{F} = \frac{\text{Anzahl unplanmäßigen Wechsels}}{\text{Teamgröße}} * 100$$

Funktionaler Umfang, gemessen in Function Points (FP), Source Lines of Code (SLOC), …
Aufwand, gemessen in Personenmonaten (PM), Personenwochen (PW), …
Dauer, gemessen in Monaten (M), Wochen M (W), …

Ausgewählte Projektmetriken 3/6

Planeinhaltung

$$PE1_{Aufwand} = \frac{IstAufwand}{Geplanter\ Aufwand}$$

>1: Planwert überschritten,
1: Plan eingehalten,
<1: Planwert unterschritten

$$PE2_{Aufwand} = \frac{IstAufwand\ - Geplanter\ Aufwand}{Geplanter\ Aufwand}$$

>0: Planwert überschritten,
0: Plan eingehalten,
<0: Planwert unterschritten

$$PE3_{Aufwand} = \frac{Abs(IstAufwand\ - Geplanter\ Aufwand)}{Geplanter\ Aufwand}$$

0: Plan eingehalten,
sonst Plan nicht eingehalten

Ähnlich wird die Planeinhaltung hinsichtlich Termin (Projektdauer) und Kosten ermittelt.

Ausgewählte Projektmetriken 4/6

Stabilität der Anforderungen

$$Änderungshäufigkeit = \frac{Anzahl\ Änderungen}{Funktionaler\ Umfang}$$

$$Änderungsgrad = \frac{Funktionaler\ Umfang\ der\ Änderungen}{Funktionaler\ Umfang} * 100$$

$$Änderungsrate = \frac{Anzahl\ Änderungen}{Anzahl\ Anforderungen} * 100$$

Funktionaler Umfang, gemessen in Function Points (FP), Source Lines of Code (SLOC), …

Ausgewählte Projektmetriken 5/6

Fehlerrate

$$FR = \frac{\text{Anzahl Fehler}}{\text{Funktionaler Umfang}}$$

Fehlerfindungsrate

$$FFR = \frac{\text{Anzahl gefundener Fehler}}{\text{Prognostizierte Gesamtzahl aller Fehler}} * 100$$

Restfehlerrate (RFR)
Von besonderem Interesse sind RFR innerhalb des ersten Monats, der ersten 3 Monate oder des ersten Jahres nach Start des Produktivbetriebs.

Funktionaler Umfang, gemessen in Function Points (FP), Source Lines of Code (SLOC), …

Ausgewählte Projektmetriken 6/6

Stabilitätsindex
Mit Stabilitätsindex wird die Produktreife (indirekt: Projektfortschritt) gemessen. Die Metrik wird auf ein Release oder eine Version einer Software angewandt.

$$S = \frac{F_{all} - (F_{new} + F_{mod} + F_{del})}{F_{all}}$$

S: Stabilitätsindex
F_{all}: Gesamtzahl der Funktionen
F_{new}: neue Funktionen
F_{mod}: modifizierte Funktionen
F_{del}: gelöschte Funktionen

Kapitel 4:
Steuerung von Softwareprojekten

4.1 Methoden der Projektsteuerung

4.2 Ausgewählte Projektmetriken

4.3 Berichterstattung

4.4 Produktivitätssteigerung

Berichterstattung

- Die Berichterstattung regelt den zielgerichteten Informationsfluss während der gesamten Projektlaufzeit.
- Ziel der Berichterstattung ist, alle Projektbeteiligten (das Projektteam, das Management und den Auftraggeber) über den aktuellen Stand des Projekts zu informieren.
- Die Berichterstattung muss den Informationsbedarf der Projektbeteiligten berücksichtigen.

Darstellung des Projektstatus

Für eine knappe Darstellung des Projektzustands verwendet die Praxis oft die Ampel-Darstellung.

Wichtig: Klare Kriterien für einzelne Farben definieren.

Die Ampel-Darstellung kann auch für einzelne Sichten verwendet werden:

- Termin
- Aufwand
- Kosten
- Funktionalität

Projekt ohne Veränderung nicht durchführbar

Korrekturmaßnahmen erforderlich

Projekt läuft wie geplant

Kapitel 4:
Steuerung von Softwareprojekten

4.1 Methoden der Projektsteuerung
4.2 Ausgewählte Projektmetriken
4.3 Berichterstattung
4.4 Produktivitätssteigerung

Produktivität

Die Steigerung der Produktivität des Entwicklungsteams ist ein zentrales Ziel der Projektleitung.

Produktivitätssteigerung bedeutet:

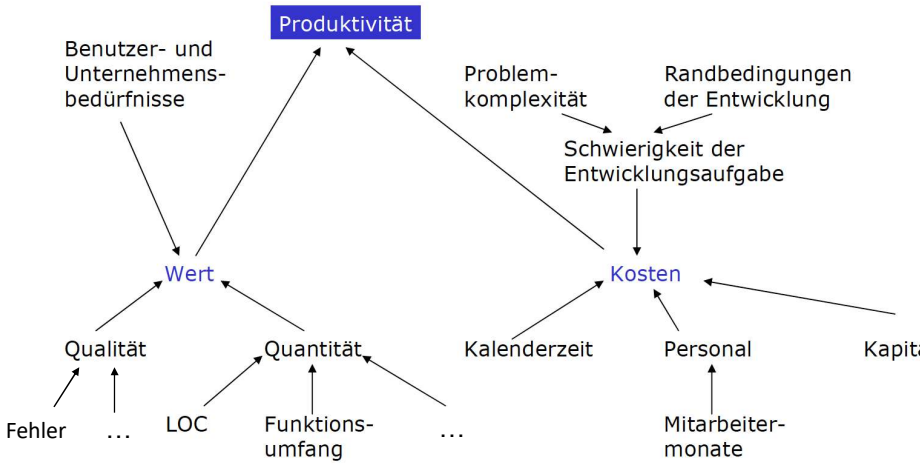
- Softwareprodukt in kürzerer Zeit entwickeln,
- Softwareprodukt mit geringerem Budget entwickeln,
- Softwareprodukt mit höherer Qualität entwickeln.

Produktivität lässt sich definieren:

- Produktivität = Leistung / Aufwand
- Produktivität = Produktwert / Kosten

Produktivität und ihre Einflussfaktoren

Es gibt zur Zeit keine befriedigende Metrik zur Messung der Produktivität. Die wesentlichen Einflussfaktoren sind aber bekannt:



Einflussfaktoren der Produktivität (i)

Produkt-/Projekteinflüsse:

- Die Produktkomplexität beeinflusst die Produktivität. Die Messung der Komplexität eines Softwareproduktes ist aber schwer.
- Produktgröße hat Einfluss auf die Entwicklungskosten, denn der Aufwand nimmt überproportional mit der Projektgröße zu.
- Geforderte Produktqualität hat Einfluss auf die Produktivität.

Prozesseinflüsse

- Einsatz geeigneter Werkzeuge und Methoden steigern die Produktivität.
- Nicht die Einführung von Werkzeugen stellt den kritischen Erfolgsfaktor dar, sondern die Einführung der Methoden, denn diese verlangen neue Denkweise von den Mitarbeitern.

Einflussfaktoren der Produktivität (ii)

Mitarbeitereinflüsse

- Programmiererfahrung und Kenntnis des Anwendungsgebiets haben Einfluss auf Leistung.

Managementeinflüsse

- Das Management hat Einfluss auf Mitarbeiter und Entwicklungsprozess.
- Arbeitsplatzausstattung/-gestaltung hat Einfluss auf die Produktivität. Ein Mitarbeiter ist umso produktiver je ruhiger und größer sein Arbeitsplatz ist, weniger er gestört wird und besser seine Privatsphäre gewahrt bleibt.
- Die Firmenkultur beeinflusst indirekt die Produktivität.
- Mit wachsender Teamgröße sinkt die Produktivität.

Produktivität und Qualität

Produktivität und Qualität beeinflussen sich. Es gibt zwei konträre Meinungen:

- Hohe Anforderungen an Qualität verringern die Produktivität.
- Hohe Anforderungen an Qualität verbessern die Produktivität.

Bedenke: 2/3 aller Lebenszykluskosten entfallen auf die Wartung und Pflege und nur 1/3 auf die eigentliche Entwicklung.

Maßnahmen zur Steigerung der Produktivität des Entwicklungsteams

- Leistung des Teams erhöhen
  - Qualifiziertes Personal einsetzen
  - Optimale Arbeitsumgebung schaffen
  - Team optimal führen und fördern
- Arbeitsschritte effizienter machen
  - CASE-Umgebungen einsetzen
  - Kommunikation im Team optimieren
- Arbeitsschritte eliminieren
  - Generatoren einsetzen
  - Qualitätssicherung automatisieren
- Überarbeitung minimieren
  - Frühe Phasen systematisieren
  - Modelle animieren/simulieren
  - Prototypen verwenden
- Komplexe Produkte in Teile zerlegen
  - Wirtschaftlichkeit analysieren
  - Inkrementell/Iterativ entwickeln
- Komponenten wiederverwenden
  - Komponentenbibliotheken anlegen
  - Wiederverwendung belohnen

Earned Value-Analyse (Aufwand-Termin-Treue) Beispiel

Projekt: Entwicklung einer Webanwendung

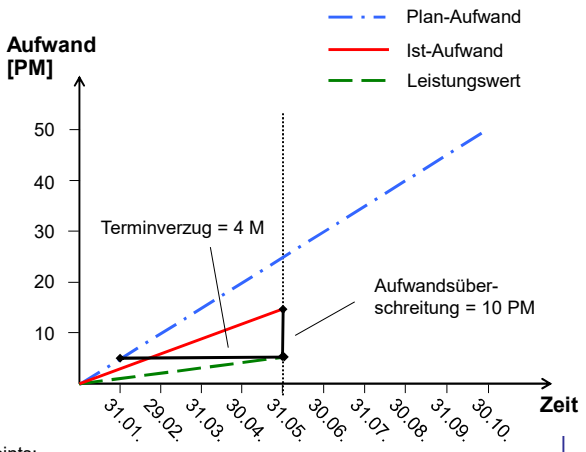
Plandaten:
Projektumfang = 1000 FP →
Aufwand = 50 PM

Dauer = 2,5 * 50^0,35 = 9,83 M
Teamgröße = 50 / 9,83 = 5,08 P

Teamgröße = 5 → Dauer = 10 M

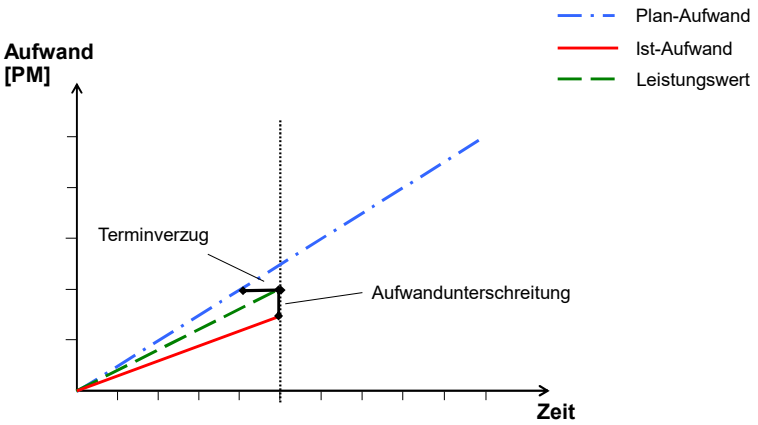
Istdaten am 31.05.:
15 PM erbracht; 100 FP implementiert.

Wie viel Aufwand wurde für die Realisierung von 100 FP geplant?
Antwort: **5 PM**

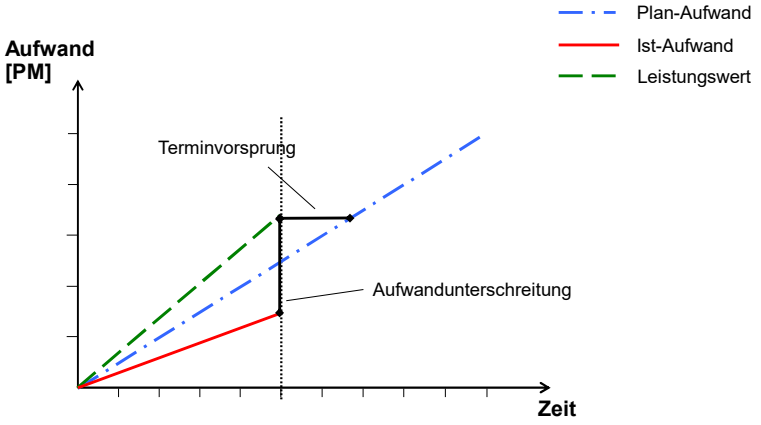


FP: Function Points;
PM: Personenmonat;
P: Person; M: Monat

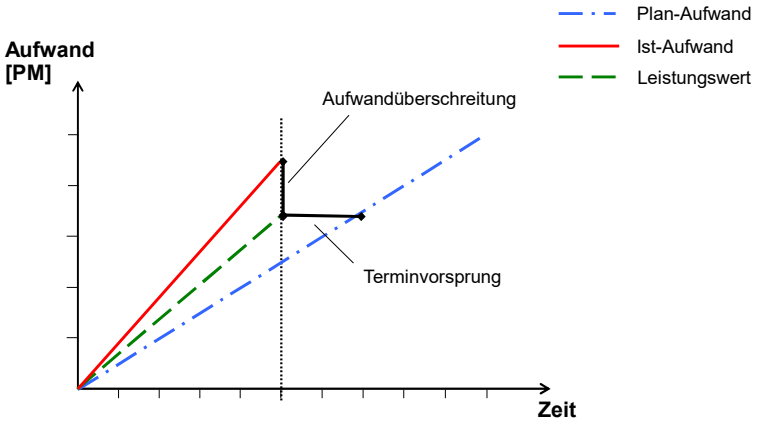
Earned Value-Analyse (Aufwand-Termin-Treue) Beispiel



Earned Value-Analyse (Aufwand-Termin-Treue) Beispiel



Earned Value-Analyse (Aufwand-Termin-Treue) Beispiel



0-100- und 0-50-100-Methode
Beispiel

	Relativer Fertigstellungsgrad*	0-100-Methode	0-50-100-Methode
AP1	0,20	0	0,50
AP2	0,30	0	0,50
AP3	1	1	1
AP4	0,40	0	0,50
AP5	1	1	1
Projekt	58%	40%	70%

* von Verantwortlichen geschätzt

Meilenstein-Trend-Analyse
Beispiel

Projektstart: 01.07.xx
M2 ist geplant für 31.10.xx

Projektverlauf:

31.07.xx: Im Plan

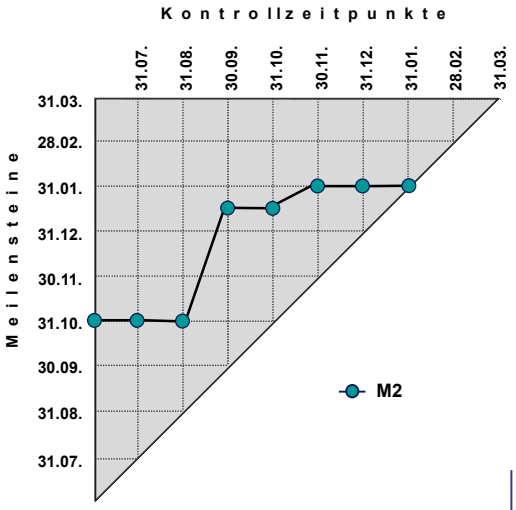
31.08.xx: Im Plan

30.09.xx: Der Termin ist nicht einzuhalten; neuer Termin: 15.01.xx

31.10.xx: Im Plan

30.11.xx: Der Termin ist nicht einzuhalten; neuer Termin: 31.01.xx

Dieser Termin wird dann eingehalten.



Stabilitätsindex
Vereinfachte Berechnung am Beispiel

$$S = \frac{F_{all} - (F_{new} + F_{mod} + F_{del})}{F_{all}}$$

Version	F_{all}	F_{new}	F_{mod}	F_{del}	S
1.0	10				
1.1	10	0	1	0	0,90
1.2	10	1	1	0	0,80
2.0	11	3	1	1	0,55
2.1	13	0	2	0	0,85

F_{all} = F_{all} der Vorversion + F_{new} der Vorversion – F_{del} der Vorversion